

What is Claimed:

1. A method of enabling a first system to use a second system comprising:
 - receiving, from the first system, a first request directed to the second system, said first request being in a form adapted for the first system but not for the second system;
 - performing a first conversion of said first request to produce a second request, said second request being in a form adapted for said second system but not for said first system;
 - invoking the processing of said second request by the second system;
 - receiving a first reply from the second system;
 - performing a second conversion of said first reply to produce a second reply; and
 - providing said second reply to said first system.
2. The method of claim 1, wherein said first request comprises a datum in a first format, and wherein said act of performing said first conversion comprises:
 - converting said datum to a second format different from said first format, said second request comprising said datum in said second format.
3. The method of claim 1, wherein said first request comprises a datum having a first type, and wherein said act of performing said first conversion comprises:
 - converting said datum to a second type different from said first type, said second request comprising said datum in said second type.
4. The method of claim 3, wherein said first type is supported in said first system but not in said second system.
5. The method of claim 3, wherein said second type differs structurally from said first type in at least one aspect.
6. The method of claim 1, wherein said first request comprises a call using a first mechanism to a software object in the second system, and wherein said act of performing said first conversion comprises:

converting said call for use with a second mechanism different from said first mechanism.

7. The method of claim 6, wherein said first mechanism comprises a commarea that is used to pass a call parameter to said object and to receive a result from said object, and wherein said second mechanism comprises:

a first area that is used to pass said call parameter, or a converted call parameter that corresponds to said call parameter, to said object; and

a second area that is used to receive said result, or a converted result that corresponds to said result, from said object.

8. The method of claim 1, wherein said first request comprises a remote call according to a first protocol, wherein said second request comprises a remote call according to a second protocol different from said first protocol, and wherein said act of performing said first conversion comprises:

preparing said second request to correspond substantively with said first request and to work in accordance with said second protocol.

9. The method of claim 8, wherein said first protocol calls for invocation to be performed with a bidirectional interaction between a caller and a callee, wherein said second protocol calls for an invocation to be performed in a unidirectional call message from said caller to said callee, and wherein said act of preparing said second request comprises:

engaging in an interaction with the caller on the first system to obtain information relating to a call;

collecting said information; and

preparing said second request using the collected information.

10. The method of claim 1, wherein said first system is adapted to communicate a remote call according to a first network protocol, and wherein said first and second requests, and said first and second replies, are transmitted using a second network protocol different from said first

network protocol, and wherein said acts of performing first and second conversions comprise:

including in said second request and said second reply header information that corresponds to information that is contained in requests or replies according to said first protocol.

11. The method of claim 1, wherein said first request comprises a call to a software object in said second system, and wherein the form of said first request is adapted for making requests from the first system to a remote system that is of the same type of environment as the first system.

12. The method of claim 1, wherein said second system comprises an environment based on a MICROSOFT WINDOWS operating system.

13. The method of claim 1, wherein said first system comprises an IBM mainframe.

14. A method of enabling a first software object in a first system to call a second software object in a second system, the method comprising:

evaluating first information that the first software object exposes when making a call to a remote system;

evaluating second information that the second software object exposes when receiving a call from a remote system;

generating conversion information descriptive of a process to be followed in order to convert the first information into a form compatible with the second information;

providing the conversion information to a conversion service that uses the conversion information to convert a first call from the first object into a call in a form usable by the second object.

15. The method of claim 14, wherein the first information comprises a call parameter in a first format, wherein the second information comprises a call parameter in a second format, and wherein the act of generating conversion information comprises:

generating code or data that describes how to convert a call parameter from the first format to the second format.

16. The method of claim 14, wherein the first information comprises a call parameter of a first data type which is not usable by the second software object, and wherein the act of generating conversion information comprises:

generating a second data type that corresponds to the first data type and which is usable by the second software object;

generating code or data that describes how to convert data of the first data type to the second data type.

17. The method of claim 14, wherein the first information comprises a return value in a first form, wherein the second information comprises a return value in a second form different from said first form, and wherein the act of generating conversion information comprises:

generating code or data that describes how to convert data in said first form to said second form.

18. The method of claim 14, wherein said first software object makes a call to a remote system according to a first programming model, wherein said second data object receives a call from a remote system according to a second programming model, and wherein the act of generating conversion information comprises:

generating code or data that indicates which programming model the first software object uses to make a remote call.

19. The method of claim 18, wherein the act of generating conversion information comprises:

generating code or data that describes at least one customization in converting from the first programming model to the second programming model.

20. The method of claim 14, wherein the act of generating conversion information comprises:

generating a transaction initiation message that is used in invoking the second software object or in reply to the first software object.

21. The method of claim 14, wherein said second system comprises an environment based on a MICROSOFT WINDOWS operating system.

22. The method of claim 14, wherein said first system comprises an IBM mainframe.

23. A system to enable a first software object in a first environment to call a second software object in a second environment, the system comprising:

a service object that receives a first request from the first software object, converts the first request into a second request which is in a form usable by the second software object, and presents the second request to the second software object.

24. The system of claim 23, wherein the service object further receives a first reply from the second software object, converts the first reply to a second reply which is in a form usable by the first software object, and provides the second reply to the first software object.

25. The system of claim 23, wherein the service object comprises:

a listener object that detects that a contact regarding the first request has been made by the first software object.

26. The system of claim 23, wherein the service object comprises:

a queuing object that queues at least one of connections and requests from the first system.

27. The system of claim 23, wherein the service object comprises:

a transit object that receives information related to the first request from the first software object and prepares the information into a form that can be used for a call to the second software object.

28. The system of claim 27, wherein the service object further comprises:

an invocation object that lays out the information prepared by the transit object into a

form that can be used for a call to the second software object, and that uses the laid out information to invoke the second software object.

29. The system of claim 23, wherein the service object comprises:

a flow control object that manages the interaction of one or more components involved in the conversion of the first request into the second request.

30. The system of claim 23, wherein the service object comprises:

an error handling object that receives an indication of an error from the second software object and packages the error into a form usable by the first system or the first software object.

31. The system of claim 30, wherein the error information comprises an FMH7 field of an SNA protocol, and wherein said error handling object creates header information representative of the contents of said FMH7 field, said header information being adapted for use with a TCP protocol.

32. A computer-readable medium encoded with computer-executable instructions to facilitate interoperability between a first system and a second system, the instructions being adapted to perform acts comprising:

receiving a first call from a first software object in the first system to a second software object in the second system, the first call being in a format that is not compatible with the second system;

converting the first call into a second call, the second call being in a format that is compatible with the second system; and

invoking the second software object using the second call.

33. The computer-readable medium of claim 32, wherein the instructions are adapted to perform acts further comprising:

listening for a connection from the first system; and
receiving information related to the first call in response to the connection.

34. The computer-readable medium of claim 32, wherein the first call comprises data in a first form, and wherein the instructions are adapted to perform acts further comprising:

converting the data from the first form into a second form usable by the second software object.

35. The computer-readable medium of claim 32, wherein the second software object provides a first reply in response to being called, and wherein the instructions are adapted to perform acts further comprising:

converting the first reply into a second reply, the second reply being in a form that is compatible with the first software object or the first system, the first reply being in a form that is not compatible with the first software object or the first system.

36. The computer-readable medium of claim 32, wherein the second software object generates error information in response to being called, and wherein the instructions are adapted to perform acts further comprising:

converting said error information into a format compatible with the first software object or the first system, or into a format compatible with a communication protocol employed by the first system.

37. The computer-readable medium of claim 32, wherein the first call is compatible with either a CICS or IMS programming model, and wherein the second call is compatible with either a COM or .NET programming model.